Asynchronous deterministic rendezvous in bounded terrains

Jurek Czyzowicz, David Ilcinkas, Arnaud Labourel, Andrzej Pelc

Asynchronous deterministic rendezvous in bounded terrains

Jurek Czyzowicz [*†] David Ilcinkas [‡] Arnaud Labourel [*§] Andrzej Pelc [*¶]

Abstract

Two mobile agents (robots) have to meet in an a priori unknown bounded terrain modeled as a polygon, possibly with polygonal obstacles. Agents are modeled as points, and each of them is equipped with a compass. Compasses of agents may be incoherent. Agents construct their routes, but the actual walk of each agent is decided by the adversary: the movement of the agent can be at arbitrary speed, the agent may sometimes stop or go back and forth, as long as the walk of the agent in each segment of its route is continuous, does not leave it and covers all of it. We consider several scenarios, depending on three factors: (1) obstacles in the terrain are present, or not, (2) compasses of both agents agree, or not, (3) agents have or do not have a map of the terrain with their positions marked. The cost of a rendezvous algorithm is the worst-case sum of lengths of the agents' trajectories until their meeting. For each scenario we design a deterministic rendezvous algorithm and analyze its cost. We also prove lower bounds on the cost of any deterministic rendezvous algorithm in each case. For all scenarios these bounds are tight.

keywords: mobile agent, rendezvous, deterministic, polygon, obstacle

[*]Département d'informatique, Université du Québec en Outaouais, Gatineau, Québec J8X 3X7, Canada. E-mails: jurek@uqo.ca, labourel.arnaud@gmail.com, pelc@uqo.ca

[†]Partially supported by NSERC discovery grant.

[‡]LaBRI, Université Bordeaux I, 33405 Talence, France. E-mail: david.ilcinkas@labri.fr

[§]This work was done during this author's stay at the Université du Québec en Outaouais as a postdoctoral fellow.

[¶]Partially supported by NSERC discovery grant and by the Research Chair in Distributed Computing at the Université du Québec en Outaouais.

1 Introduction

The problem and the model. Two mobile agents (robots) modeled as points starting at different locations of an a priori unknown bounded terrain have to meet. The terrain is represented as a polygon possibly with a finite number of polygonal obstacles. We assume that the boundary of the terrain is included in it. Thus, formally, a terrain is a set $\mathcal{P}_0 \setminus (\mathcal{P}_1 \cup \cdots \cup \mathcal{P}_k)$, where $\mathcal{P}_0$ is a closed polygon and $\mathcal{P}_1, \ldots, \mathcal{P}_k$ are disjoint open polygons included in $\mathcal{P}_0$. We assume that an agent knows if it is at an interior or at a boundary point, and in the latter case it is capable of walking along the boundary in both directions (i.e., it knows the slope(s) of the boundary at this point). However, an agent cannot sense the terrain or the other agent at any vicinity of its current location. Meeting (rendezvous) is defined as the equality of points representing agents at some moment of time.

We assume that each agent has a unit of length and a compass. Compasses of agents may be incoherent, however we assume that agents have the same (clockwise) orientation of their system of coordinates. An additional tool, which may or may not be available to the agents, is a map of the terrain. The map available to an agent is scaled (i.e., it accurately shows the distances), distinguishes the starting positions of this agent and the other one, and is oriented according to the compass of the agent. (Hence maps of different agents may have different North.)

All our considerations concern deterministic algorithms. The crucial notion is the *route* of the agent which is a finite polygonal path in the terrain. The adversary initially places an agent at some point in the terrain. The agent constructs its route in steps in the following way. In every step the agent starts at some point v; in the first step, v is the starting point chosen by the adversary. The agent chooses a direction α, according to its compass, and a distance x. If the segment of length x in direction α starting in v does not intersect the boundary of the terrain, the step ends when the agent reaches point u at distance x from v in direction α. Otherwise, the step ends at the closest point of the boundary in direction α. If the starting point v in a step is in a segment of the boundary of the terrain, the agent has also an option (in this step) to follow this segment of the boundary in any of the two directions till its end or for some given distance along it. Steps are repeated until rendezvous, or until the route of the agent is completed.

We consider the *asynchronous* version of the rendezvous problem. The asynchrony of the agents' movements is captured by the assumption that the actual walk of each agent is decided by the adversary: the movement of the agent can be at arbitrary speed, the agent may sometimes stop or go back and forth, as long as the walk of the agent in each segment of its route is continuous, does not leave it and covers all of it. More formally, the route in a terrain is a sequence $(S_1, S_2, \ldots, S_k)$ of segments, where $S_i = [a_i, a_{i+1}]$ is the segment corresponding to step i. In our algorithms the route is always finite. This means that the agent stops at some point, regardless of the moves of the other agent. We now describe the walk f of an agent on its route. Let $R = (S_1, S_2, \ldots, S_k)$ be the route of an agent. Let $(t_1, t_2, \ldots, t_{k+1})$, where $t_1 = 0$, be an increasing sequence of reals, chosen by the adversary, that represent points in time. Let $f_i : [t_i, t_{i+1}] \to [a_i, a_{i+1}]$ be any continuous function, chosen by the adversary, such that $f_i(t_i) = a_i$ and $f_i(t_{i+1}) = a_{i+1}$. For any $t \in [t_i, t_{i+1}]$, we define $f(t) = f_i(t)$. The interpretation of the walk f is as follows: at time t the agent is at the point $f(t)$ of its route and after time t_{k+1} the agent remains inert. This general definition of the

walk and the fact that it is constructed by the adversary capture the asynchronous characteristics of the process. Throughout the paper, *rendezvous* means deterministic asynchronous rendezvous.

Agents with routes R_1 and R_2 and with walks f_1 and f_2 meet at time t, if points $f_1(t)$ and $f_2(t)$ are equal. A rendezvous is guaranteed for routes R_1 and R_2, if the agents using these routes meet at some time t, regardless of the walks chosen by the adversary. The trajectory of an agent is the sequence of segments on its route until rendezvous. (The last segment of the trajectory of an agent may be either the last segment of its route or any of its segments or a portion of it, if the other agent is met there.) The cost of a rendezvous algorithm is the worst case sum of lengths of segments of trajectories of both agents, where the worst case is taken over all terrains with the considered values of parameters, and all adversarial decisions.

We consider several scenarios, depending on three factors: (1) obstacles in the terrain are present, or not, (2) compasses of both agents agree, or not, (3) agents have or do not have a map of the terrain. Combinations of the presence or absence of these factors give rise to eight scenarios. For each scenario we design a deterministic rendezvous algorithm and analyze its cost. We also prove lower bounds on the cost of any deterministic rendezvous algorithm in each case. For all scenarios these bounds are tight.

One final precision has to be made. For all scenarios except those with incoherent compasses and the presence of obstacles (regardless of the availability of a map), agents may be anonymous, i.e., they execute identical algorithms. By contrast, with the presence of obstacles and incoherent compasses, anonymity would preclude feasibility of rendezvous in some situations. Consider a square with one square obstacle positioned at its center. Consider two agents starting at opposite (diagonal) corners of the larger square, with compasses pointing to opposite North directions. If they execute identical algorithms and walk at the same speed, then at each time they are in symmetric positions in the terrain and hence rendezvous is impossible. The only way to break symmetry for a deterministic rendezvous in this case is to equip the agents with distinct labels (which are positive integers). Hence, this is the assumption we make for the scenarios with the presence of obstacles and incoherent compasses (both with and without a map). For any label μ, we denote by $|\mu|$ the length of the binary representation of the label, i.e., $|\mu| = \lfloor \log \mu \rfloor + 1$.

Our results. The cost of our algorithms depends on some of the following parameters (different parameters for different scenarios, see the discussion in Section 4): D is the distance between starting positions of agents in the terrain (i.e., the length of the shortest path between them included in the terrain), P is the perimeter of the terrain, (i.e., the sum of perimeters of all polygons $\mathcal{P}_0, \mathcal{P}_1, \ldots, \mathcal{P}_k$), x is the largest perimeter of an obstacle, and l and L are the smaller and larger labels of agents, respectively, for the two scenarios that require different labels, as remarked above., i.e., for the scenarios with the presence of obstacles and incoherent compasses.

Our rendezvous algorithms rely on two different ideas: either meeting in a uniquely defined point of the terrain, or meeting on a uniquely defined cycle. It turns out that a uniquely defined point can be found in all scenarios except those with the presence of obstacles and incoherent compasses. In this case even anonymous agents can meet. On the other hand, with the presence of obstacles and incoherent compasses, such a uniquely defined point may not exist, as witnessed by the above

quoted example of a square with one square obstacle positioned at its center. For these scenarios we resort to the technique of meeting at a common cycle, breaking symmetry by different labels of agents.

We first summarize our results concerning rendezvous when each of the agents is equipped with a map showing its own position and that of the other agent. If compasses of the agents are coherent, then we show a rendezvous algorithm at cost D, which is clearly optimal. Otherwise, and if the terrain does not contain obstacles, then we show an algorithm whose cost is again D, and hence optimal. Finally, with incoherent compasses in the presence of obstacles, we show a rendezvous algorithm at cost $O(D|l|)$; in the latter scenario we show that cost $\Omega(D|l|)$ is necessary for some terrains.

Our results concerning rendezvous without a map are as follows. If compasses of the agents are coherent, then we show a rendezvous algorithm at cost $O(P)$. We also show a matching lower bound $\Omega(P)$ in this case. If compasses of the agents are incoherent, but the terrain does not contain obstacles, then we show a rendezvous algorithm at cost $O(P)$ and again a matching lower bound $\Omega(P)$. Finally, in the hardest of all scenarios (presence of obstacles, incoherent compasses and no map) we have a rendezvous algorithm at cost $O(P + x|L|)$ and a matching lower bound $\Omega(P + x|L|)$. Table 1 summarizes our results.

Rendezvous with a map			Rendezvous without a map		
obstacles \ compasses	coherent	incoherent	obstacles \ compasses	coherent	incoherent
no	D	D	no	$\Theta(P)$	$\Theta(P)$
yes		$\Theta(D\|l\|)$	yes		$\Theta(P + x\|L\|)$

Table 1: Summary of results

Related work. The rendezvous problem was first described in [30]. A detailed discussion of the large literature on rendezvous can be found in the excellent book [4]. Most of the results in this domain can be divided into two classes: those considering the geometric scenario (rendezvous in the line, see, e.g., [10, 11, 20, 31], or in the plane, see, e.g., [7, 8]), and those discussing rendezvous in graphs, e.g., [2, 5]. Some of the authors, e.g., [2, 3, 6, 10, 22] consider the probabilistic scenario where inputs and/or rendezvous strategies are random. Randomized rendezvous strategies use random walks in graphs, which were thoroughly investigated and applied also to other problems, such as graph traversing [1], on-line algorithms [15] and estimating volumes of convex bodies [18]. A generalization of the rendezvous problem is that of gathering [19, 22, 23, 24, 27, 32], when more than two agents have to meet in one location.

If graphs are unlabeled, deterministic rendezvous requires breaking symmetry, which can be accomplished either by allowing marking nodes or by labeling the agents. Deterministic rendezvous with anonymous agents working in unlabeled graphs but equipped with tokens used to mark nodes was considered e.g., in [26]. In [33] the authors studied the task of gathering many agents with unique labels. In [17, 25, 34] deterministic rendezvous in graphs with labeled agents was considered. However, in all the above papers, the synchronous setting was assumed. Asynchronous gathering under geometric scenarios has been studied, e.g., in [13, 19, 28] in different models than ours: agents

could not remember past events, but they were assumed to have at least partial visibility of the scene. The first paper to consider deterministic asynchronous rendezvous in graphs was [14]. The authors concentrated on complexity of rendezvous in simple graphs, such as the ring and the infinite line. They also showed feasibility of deterministic asynchronous rendezvous in arbitrary finite connected graphs with *known* upper bound on the size. Further improvements of the above results for the infinite line were proposed in [31]. Gathering many robots in a graph, under a different asynchronous model and assuming that the whole graph is seen by each robot, has been studied in [23, 24].

2 Rendezvous with a map

We start by describing the following procedure that finds a unique shortest path from the starting position of one agent to the other. The procedure works in all scenarios in which agents have a map of the terrain with their positions indicated.

Procedure path `UniquePath`(point v, point w)
1 point $u := v$; path $p := \{v\}$;
2 $\mathcal{S} = \{p_s \mid p_s$ is a shortest path between v and $w\}$;
3 **while** $(u \neq w)$ **do**
4 $\mathcal{U}$:=all paths p_s of $\mathcal{S}$ such that the first segment of the subpath of p_s leading from u to w is the first in clockwise order around u starting from the direction vw;
5 $p' := \bigcap_{p_s \in \mathcal{U}} p_s$;
6 extend p with the connected part of p' containing u;
7 $u :=$ new end of path p;
8 **return** p;

Lemma 2.1 *Procedure* `UniquePath` *computes a unique shortest path from v to w, independent of the agent computing it.*

Proof: All shortest paths between two points inside a terrain can be computed as in [21]. The path computed by the call of `UniquePath`(v, w) is a shortest path, since it is composed, by construction, of parts of shortest paths between v and w. The path is computed in a deterministic way without using the compass direction of the agent or the unit of length of the agent. Hence, it is unique. □

2.1 Coherent compasses

If agents have a map and coherent compasses, then they can easily agree on one of their two starting positions and meet at this point at cost D, which is optimal. This is done by the following Algorithm RVCM (rendezvous with a map and coherent compasses).

Algorithm *RVCM*
Let v be the northernmost of the two starting positions of the agents. If both agents have the same latitude, let v be the easternmost of them. Let w be the other starting position. The agent starting at v remains inert. The agent starting at w computes the path $p = \texttt{UniquePath}(w, v)$ and moves along p until v.

Theorem 2.1 *Algorithm RVCM guarantees rendezvous at cost D, for any two agents with a map and coherent compasses, in any terrain.*

Proof: The position v computed by the two agents is the same, since they have coherent compasses. The agents will eventually meet in v. The cost of rendezvous is D, since p is of length D. □

2.2 Incoherent compasses

2.2.1 Terrains without obstacles

In an empty polygon there is a unique shortest path between starting positions of the agents [9], and agents with a map can meet in the middle of this path at cost D, which is optimal. This is done by Algorithm RVM (rendezvous with a map, without obstacles).

Algorithm *RVM*
The agent computes the (unique) shortest path between the starting positions of the two agents. Then, it moves along this shortest path until the middle of it.

Theorem 2.2 *Algorithm RVM guarantees rendezvous at cost D for any two agents with a map, in any terrain without obstacles.*

Proof: In a polygon without obstacles, the shortest path between two points is unique and can be computed as in [21]. The two agents will eventually meet in the middle of this shortest path. The cost of rendezvous is D, since the path is of length D. □

2.2.2 Terrains with obstacles

This is the first of the two scenarios where agents cannot always predetermine a meeting point. Therefore they compute a common embedding of a ring on which they are initially situated, and then each agent executes the rendezvous algorithm from [14] for this ring. Rendezvous is guaranteed to occur on the ring, but the meeting point depends on the walks of the agents determined by the adversary. This is done by Algorithm RVMO (rendezvous with a map, with obstacles).

Algorithm *RVMO*

Phase 1: computation of the embedding[*] $\mathcal{R}$ **of a ring of size** 4**.**
Let v be the starting position of the agent and let w be the starting position of the other agent. The agent computes the embedding $\mathcal{R}$ of a ring, composed of four nodes v, a, w and b, where a is the midpoint of `UniquePath`(v, w), b is the midpoint of `UniquePath`(w, v), and the four edges are the respective halves of these paths.
Phase 2: rendezvous on $\mathcal{R}$**.**
This phase consists in applying the rendezvous algorithm from [14] for ring $\mathcal{R}$, whose size (four) is known to the agents.

[a] This embedding is not necessarily homeomorphic with a circle, it may be degenerated.

Theorem 2.3 *Algorithm RVMO guarantees rendezvous at cost $O(D|l|)$ for arbitrary two agents with a map, in any terrain.*

Proof: Let a_1 and a_2 be the two agents that have to meet. The embedding $\mathcal{R}$ of the ring is the same for the two agents by Lemma 2.1. The algorithm from [14] guarantees rendezvous and has complexity expressed in terms of the total number of edge traversals by the agents before rendezvous occurs, equal to $O(n|l|)$, where n is the number of nodes of the ring and l is the smaller of the two labels of the agents. Since the ring has size four and each of its edges has length $D/2$, the total cost of rendezvous is $O(D|l|)$. □

The following lower bound shows that the cost of Algorithm RVMO cannot be improved for some terrains. Indeed, it implies that for all $D > 0$, there exists a polygon with a single obstacle, for which the cost of any rendezvous algorithm for two agents, starting at distance D, is $\Omega(D|l|)$.

Theorem 2.4 *For any rendezvous algorithm A, for any $D > 0$, and for any integers $k_2 \geq k_1 > 0$, there exist two labels l_1 and l_2 of lengths at most k_1 and at most k_2, respectively, and a polygon with a single obstacle of perimeter $2D$, such that algorithm A executed by agents with labels l_1 and l_2 starting at distance D, requires cost $\Omega(Dk_1)$. This holds even if the two agents have a map.*

Proof: The idea of the proof is based on an argument from [17]. For $y > 0$, we consider a terrain $\mathcal{T}$ that is a hexagon of side $y + 2$ with one hexagonal obstacle of side y with the same center. The two agents start at positions u and v in $\mathcal{T}$, as depicted in Figure 1. The compasses of agents point in opposite directions. Observe that $D = 3y$. We call *slices* the six trapezoids bounded by two corresponding parallel sides of the two hexagons and by the segments linking the corresponding vertices of the hexagons. To avoid ambiguity, we say that an agent in the segment shared by two slices is in the first of them in clockwise order. Note that agents start in two different slices with two slices in between.

Fix a rendezvous algorithm A. We assume that both agents always move at the same constant speed. We divide the execution of algorithm A into periods during which each agent traverses a distance y. During any period, an agent can only visit the slice where it starts the period and one

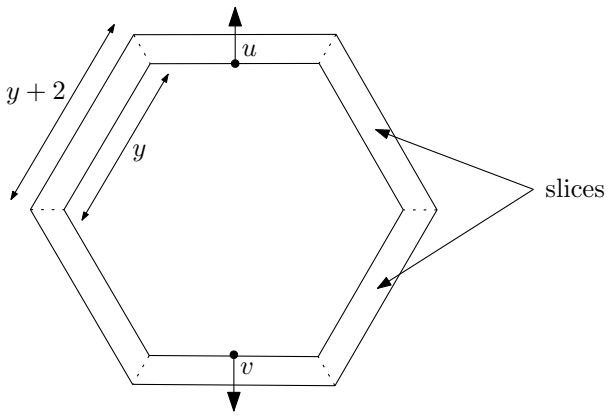

Figure 1: Terrain $\mathcal{T}$

of the two adjacent slices. The behavior of an agent with label l, running algorithm A, yields the following sequence of integers from the set $\{-1, 0, 1\}$, called the behavior code. The i-th term of the behavior code of an agent is -1 if the agent ends period i in the slice preceding (in clockwise order) the slice in which it began the period, 1 if it ends period i in the slice following it (in clockwise order), and 0 if it begins and ends period i in the same slice. Due to the symmetry of the figure and to opposite compasses an agent with a given label has the same behavior code if it starts at point u or at point v. Note that two agents with the same prefix of length k_1 of their behavior codes cannot accomplish rendezvous during the first k_1 periods, since they start separated by at least two slices, and they cannot be in the same slice during any period.

There are less than $3^{k_1/2} < 2^{k_1}$ behavior codes of length at most $k_1/2$. Hence it is possible to pick two distinct labels l_1 and l_2 of lengths at most k_1, respectively, such that the prefix of length $k_1/2$ of their behavior codes is the same. For these labels, algorithm A does not accomplish rendezvous before both agents have travelled a distance $yk_1/2 = \Omega(Dk_1)$. □

3 Rendezvous without a map

3.1 Coherent compasses

It turns out that agents can recognize the outer boundary of the terrain even without a map. Hence, if their compasses are coherent, they can identify a uniquely defined point on this boundary and meet in this point. This is done by Algorithm RVC (rendezvous with coherent compasses) at cost $O(P)$.

Algorithm *RVC*

From its starting position v, the agent follows the half-line α pointing to the North, as far as possible. When it hits the boundary of a polygon $\mathcal{P}$ (i.e., either the external boundary of the terrain or the boundary of an obstacle), it traverses the entire boundary of $\mathcal{P}$. Then, it computes the point u which is the farthest point from v in $\mathcal{P} \cap \alpha$. It goes around $\mathcal{P}$ until reaching u again and progresses on α, if possible. If this is impossible, the agent recognizes that it went around the boundary of $\mathcal{P}_0$. It then computes the northernmost points in $\mathcal{P}_0$. Finally, it traverses the boundary of $\mathcal{P}_0$ until reaching the easternmost of these points.

Theorem 3.1 *Algorithm RVC guarantees rendezvous at cost $O(P)$ for any two agents with coherent compasses, in any terrain.*

Proof: The first phase of the algorithm that consists in reaching $\mathcal{P}_0$ and making the tour of the boundary of $\mathcal{P}_0$ costs at most $3P$, since the boundary of each polygon of the terrain is traversed at most twice and the total length of parts of α inside the terrain is at most P. Reaching the rendezvous point costs at most P. The agents will eventually meet in the easternmost of the northernmost points of $\mathcal{P}_0$, since they have coherent compasses and this point is unique. □

The following lower bound shows that the cost of Algorithm RVC is asymptotically optimal, for some polygons even without obstacles. This lower bound $\Omega(P)$ holds even if the distance D between starting positions of agents is bounded and if their compasses are coherent.

Theorem 3.2 *There exists a polygon of an arbitrarily large perimeter P, for which the cost of any rendezvous algorithm for two agents with coherent compasses starting at any distance $D > 0$, is $\Omega(P)$.*

Proof: Consider the polygon $\mathcal{P}'$ obtained by attaching to each side of a regular k-gon, whose center is at distance $D/8$ from its boundary, a rectangle of length $3D/8$ and of height equal to the side length of the k-gon. The polygon $\mathcal{P}$ is the polygon obtained by gluing two copies of $\mathcal{P}'$ by the small side of one of the rectangles, as depicted in Fig. 2. Let P be the perimeter of the polygon $\mathcal{P}$. We choose $k = \Theta(P/D)$. There are two types of rectangles in $\mathcal{P}$, two *passing* ones (they share one side) and the $2k - 2$ *normal* ones.

Consider all rotations of the polygon $\mathcal{P}$ around its center of symmetry by angles $2\pi i/k$, for $i = 0, \ldots, k-1$. We will prove that any deterministic rendezvous algorithm requires cost $\Omega(P)$ in at least one of the rotated polygons. Each agent starts in the center of a different k-gon. We say that an agent has *penetrated* a rectangle if it has moved at distance $D/8$ inside the rectangle. In order to accomplish rendezvous, at least one agent has to penetrate a passing rectangle. Each time one agent penetrates a rectangle, the adversary chooses a rotation, so that all previously penetrated rectangles, including the current one, are normal rectangles. This choice is coherent with the knowledge previously acquired by the agents, since normal rectangles are undistinguishable from each other and an agent needs to penetrate a rectangle in order to distinguish its type. Hence,

the two agents have to penetrate a total of $k-1$ rectangles before the adversary cannot rotate the figure to prevent the penetration of a passing rectangle. It follows that at least one of the agents has to traverse a total distance of $\Omega(kD) = \Omega(P)$ before meeting.

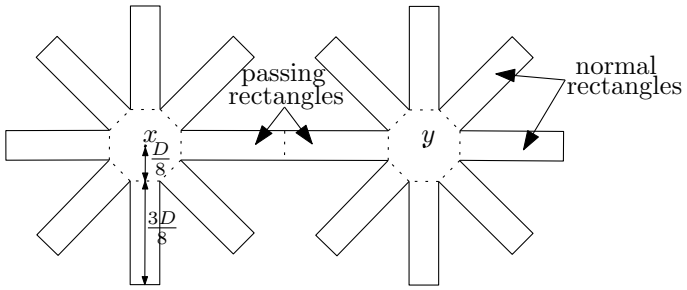

Figure 2: Polygon $\mathcal{P}$

□

3.2 Incoherent compasses

3.2.1 Terrains without obstacles

In this section, we use the notion of medial axis, proposed by Blum [12], to define a unique point of rendezvous inside the terrain. Observe that we cannot use the centroid for the rendezvous point since, as we also consider non-convex terrains, the centroid is not necessarily inside the terrain. The *medial axis* $M(\mathcal{P})$ of a polygon $\mathcal{P}$ is defined as the set of points inside $\mathcal{P}$ which have more than one closest point on the boundary of $\mathcal{P}$. Actually, $M(\mathcal{P})$ is a planar tree contained in $\mathcal{P}$, in which nodes are linked by either straight-line segment or arcs of parabolas [29]. We define the *medial point* of a polygon $\mathcal{P}$ as either the central node of $M(\mathcal{P})$ or the middle of the central edge of $M(\mathcal{P})$, depending on whether $M(\mathcal{P})$ has a central node or a central edge. Remark that the medial point of $\mathcal{P}$ is unique and is inside $\mathcal{P}$. The medial axis of a polygon $\mathcal{P}$ can be computed as in [16]. Algorithm RV (rendezvous without obstacles, without a map and with possibly incoherent compasses) determines the unknown (empty) polygon and guarantees meeting in its medial point.

Algorithm *RV*

At its starting position, the agent chooses an arbitrary half-line α which it follows until it hits the boundary of the polygon $\mathcal{P}_0$. It traverses the entire boundary of $\mathcal{P}_0$ and computes the medial point v of $\mathcal{P}_0$. Then, it moves to v by a shortest path and stops.

Theorem 3.3 *Algorithm RV guarantees rendezvous at cost $O(P)$ for any two agents, in any terrain without obstacles.*

Proof: The cost of reaching the boundary of $\mathcal{P}$ and completing a tour of it is at most $2P$. The agent can compute the medial point of the polygon and reach it at cost at most P. The two agents will eventually meet at the medial point, since it is unique. □

The lower bound from Theorem 3.2 shows that the cost of Algorithm RV cannot be improved for some polygons.

3.2.2 Terrains with obstacles

Our last rendezvous algorithm, Algorithm RVO, works for the hardest of all scenarios: rendezvous with obstacles, no map, and possibly incoherent compasses. Here again it may be impossible to predetermine a meeting point. Thus agents identify a common cycle and meet on this cycle. The difference between the present setting and that of Algorithm $RVMO$, where a map was available, is that now agents may start outside of the common cycle and have to reach it before attempting rendezvous on it. Also the common cycle is different: rather than being composed of two shortest paths between initial positions of the agents (a map seems to be needed to find such paths), it is the boundary of a (possible) obstacle $\mathcal{O}$ in which the medial point of the outer polygon is hidden. These changes have consequences for the cost of the algorithm. The fact that the medial point of the outer polygon has to be found and the obstacle $\mathcal{O}$ has to be reached is responsible for the summand P in the cost. The only bound on the perimeter of this obstacle is x. Finally, the fact that the adversary may delay the agent with the smaller label and force the other agent to make its tours of obstacle $\mathcal{O}$ before the agent with the smaller label even reaches the obstacle, is responsible for the summand $x|L|$, rather than $x|l|$, in the cost.

A *cycle* is a polygonal path whose both extremities are the same point. A *tour* of a cycle $\mathcal{C}$ is any sequence of all the segments of $\mathcal{C}$ in either clockwise or counterclockwise order starting from a vertex of $\mathcal{C}$. By extension, a *partial tour* of $\mathcal{C}$ is a path which is a subsequence of a tour of $\mathcal{C}$ with the first or the last segment of the subsequence possibly replaced by a subsegment of it.

Algorithm *RVO*

Phase 1: Computation of the medial point of $\mathcal{P}_0$
At its starting position z, the agent chooses an arbitrary half-line α which it follows as far as possible. When it hits the boundary of a polygon $\mathcal{P}$, it traverses the entire boundary of $\mathcal{P}$. Then, it computes the point w which is the farthest point from z in $\mathcal{P} \cap \alpha$. It goes around $\mathcal{P}$ until reaching w again and progresses on α, if possible. If this is impossible, the agent recognizes that it went around the boundary of $\mathcal{P}_0$. The agent computes the medial point v of $\mathcal{P}_0$.
Phase 2: Moving to the medial point of $\mathcal{P}_0$
Let u be the current position of the agent. The agent follows the segment $\overline{uv}$ as far as possible. Similarly as in the first phase of the algorithm, if the agent hits a polygon $\mathcal{P}$, it traverses the entire boundary of $\mathcal{P}$. Then, it computes the point w which is the farthest point from u in $\mathcal{P} \cap \overline{uv}$. It goes around $\mathcal{P}$ until reaching w again and progresses on α, if possible. If this is impossible and if the point v has not been reached, the agent recognizes that the point v is inside an obstacle $\mathcal{O}$, and executes phase 3. If the agent reaches v, it does not enter phase 3 of the algorithm and stops.
Phase 3: Rendezvous around the medial obstacle of the terrain
The agent goes around the obstacle $\mathcal{O}$ until it reaches a vertex s. The agent produces the modified label μ^* consisting of the binary representation of the label μ of the agent followed by a 1 and then followed by $|\mu|$ zeros. This phase consists of $|\mu^*|$ stages. In stage i, the agent completes two tours of the boundary of $\mathcal{O}$, starting and ending in s, clockwise if the i-th bit of μ^* is 1 and counterclockwise otherwise.

Let $\overline{u_1u_2}$ and $\overline{u_2u_3}$ be consecutive segments in clockwise order (resp. counterclockwise order) of a cycle. For a given walk f of an agent a, we say that the agent *traverses in a clockwise way* (resp. *in a counterclockwise way*) a vertex u_2 of a cycle at time t if $f(t) = u_2$ and there exist positive reals ϵ_1 and ϵ_2 and points y and z such that $y = f(t - \epsilon_1)$ is an internal point of $\overline{u_1u_2}$, $z = f(t + \epsilon_2)$ is an internal point of $\overline{u_2u_3}$ and the agent walks in $\overline{u_1u_2} \cup \overline{u_2u_3}$ during the time period $[t - \epsilon_1, t + \epsilon_2]$.

Before establishing the correctness and cost of Algorithm *RVO*, we need to show the following two lemmas.

Lemma 3.1 *Consider two agents on cycle $\mathcal{C}$. Suppose that one agent executes a tour of $\mathcal{C}$ in some sense of rotation, starting and ending in v. If during the same period of time, the other agent either traverses v for the first time in the other sense of rotation or does not traverse it at all, then the two agents meet.*

Proof: Let f_1 and f_2 be the walks of agents a_1 and a_2, respectively. Let t' be the moment when agent a_1 starts its tour of $\mathcal{C}$ at some vertex v. Let t'' be the moment when agent a_1 ends its tour, if agent a_2 does not traverse v in the same period of time, or, otherwise, the first moment after t' when agent a_2 traverses v. We cut cycle $\mathcal{C}$ at vertex v obtaining the path p with extremities v' and v'' that are copies of v. The walks f_1 and f_2, during the time period $[t', t'']$, can be transposed in p, since neither of the two agents traverses v during the period (t', t''). For any $t \in [t', t'']$, let $d_i(t)$ be the distance of agent a_i from v' at time t, counted on p. The two functions d_1 and d_2

are continuous, since the walks of both agents on p are continuous. Notice that, since the first traversal of v by agent a_2 may be only in the sense of rotation opposite to that of agent a_1, we have $f_1(t') = v'$ and either $f_2(t'') = v'$ or $f_1(t'') = v''$. Let $\delta(t) = d_1(t) - d_2(t)$. We have $\delta(t') = d' \leq 0$ and $\delta(t'') = d'' \geq 0$, since $d_1(t') = 0$ and $d_1(t'') \geq d_2(t'')$. The function δ is thus a continuous function from the interval $[t', t'']$ onto some interval $[c', c'']$, where $c' \leq d'$ and $c'' \geq d''$. Since 0 belongs to the interval $[c', c'']$, there must exist a moment t in the interval $[t', t'']$, for which $\delta(t) = 0$. For this moment, $f_1(t) = f_2(t)$ and the rendezvous occurs. □

Lemma 3.2 *Consider two agents on a cycle $\mathcal{C}$ and let $k \geq 0$ be an integer. If an agent executes either a partial tour of $\mathcal{C}$ followed by at most k tours of $\mathcal{C}$, or at most k tours of $\mathcal{C}$ followed by a partial tour of $\mathcal{C}$, while the second agent executes $k+2$ tours of $\mathcal{C}$, then the two agents meet.*

Proof: Assume for contradiction that the two agents never meet. During each tour of $\mathcal{C}$ by the second agent, the first agent has to traverse the starting position v of the second agent, in view of Lemma 3.1. Hence, the first agent has traversed $k+2$ times vertex v. Notice that an agent cannot traverse v without executing a tour of $\mathcal{C}$ as v is an extremity of a segment of its route. Hence the first agent has completed at least $k+1$ complete tours of $\mathcal{C}$ starting and ending at v. Finally, the first agent has started executing its tours at point v, a contradiction. □

Theorem 3.4 *Algorithm RVO guarantees rendezvous at cost $O(P + x|L|)$ for any two agents in any terrain for which x is the largest perimeter of an obstacle.*

Proof: Let a_1 and a_2 be the two agents that have to meet. The first phase of the algorithm that consists in reaching $\mathcal{P}_0$ and making the tour of the boundary of $\mathcal{P}_0$ costs at most $3P$, since the boundary of each polygon of the terrain is traversed at most twice and the total length of parts of α inside the terrain is at most P. For the same reason as in phase 1, the total cost of phase 2 is at most $3P$.

If the medial point of $\mathcal{P}_0$ is inside the terrain, then the agents meet at the end of phase 2 at total cost of at most $12P$. Otherwise, both agents eventually enter phase 3 of the algorithm and they are on the boundary of the obstacle $\mathcal{O}$ containing the medial point of $\mathcal{P}_0$. The cost follows from the fact that each agent travels a distance $O(x|L|)$ in phase 3. Indeed, each agent executes at most $2|L| + 1$ stages and each stage costs at most $2x$. Hence it remains to show that rendezvous occurs in this case as well.

Assume for contradiction that the two agents never meet. Notice that the modified label l^* cannot be the suffix of the modified label L^*. Indeed, if $|l^*| = |L^*|$ then the two labels are different since $l \neq L$, and otherwise the second part of l^*, consisting of 1 followed by $|l|$ zeros, cannot be the suffix of L^*. Hence, there exists an index i such that the $(|l^*| - i)$-th bit of l^* differs from the $(|L^*| - i)$-th bit of L^*. We call *important* stages the $(|l^*| - i)$-th stage of the agent with label l and the $(|L^*| - i)$-th stage of the agent with label L.

For $j = 1, 2$, let t_j be the moment when agent a_j enters its important stage and let t' be the first moment when both agents have finished the execution of the algorithm. Suppose by symmetry

that $t_1 \leq t_2$, i.e., agent a_1 was the first to enter its important stage. Then a_2 must have entered its important stage during the first tour of the important stage of a_1. Otherwise, agent a_2 would have completed $2i+2$ tours between t_2 and t', while agent a_1 would have completed at most $2i+1$ tours. Hence, the two agents would have met in view of Lemma 3.2. Hence, from the time t_2, agent a_2 completes one tour in some sense of rotation, starting and ending at a vertex v, while agent a_1 either traverses v for the first time in the other sense of rotation or does not traverse it at all. Hence by Lemma 3.1, the two agents meet. □

The following result gives a lower bound matching the cost of Algorithm RVO.

Theorem 3.5 *There exist terrains for which the cost of any rendezvous algorithm is* $\Omega(P + x|L|)$. *This holds for arbitrarily small* $D > 0$.

Proof: Since our lower bound is expressed as a sum, in order to prove it, we show two examples, one in which the first summand is as small as possible and the bound is equal to the other summand, and vice-versa. The first example uses the polygon from Theorem 2.4: P must be at least x and in this example we have $P = \Theta(x)$ and the lower bound is $\Omega(x|L|)$. Indeed, consider two integers $m_2 \leq m_1$. By Theorem 2.4, applied for $k_1 = k_2 = m_1$, and for any rendezvous algorithm A, there exists a label L of length m_1, such that the sum of lengths of segments of the route produced by the execution of A by an agent a_1 with label L is $\Omega(xm_1)$. The adversary chooses as the initial position of the second agent a_2 any point outside a path p of length $\Theta(xm_1)$, which is a prefix of the route of agent a_1. This point can be chosen arbitrarily close to the initial position of the first agent. The label of agent a_2 is of length m_2. Suppose that the start of agent a_2 is delayed by the adversary and occurs when p is entirely traversed by agent a_1. The two agents do not meet during this traversal of p by the first agent and so the cost of rendezvous is $\Omega(xm_1) = \Omega(x|L|)$. The second example is given by the proof of Theorem 3.2. Indeed, in this example there are no obstacles and hence $x = x|L| = 0$, while the lower bound is $\Omega(P)$. □

4 Discussion of parameters

We presented rendezvous algorithms, analyzed their cost and proved matching lower bounds in all considered scenarios. However, it is important to note that the formulas describing the cost depend on the chosen parameters in each case. All our results have the following form. For a given scenario we choose some parameters (among D, P, x, l, L), show an algorithm whose cost in any terrain is $O(f)$, where f is some simple function of the chosen parameters, and then prove that for some class of terrains any rendezvous algorithm requires cost $\Omega(f)$, which shows that the complexity of our algorithm cannot be improved in general, for the chosen parameters.

This yields the question which parameters should be chosen. In the case of complexities D and $\Theta(P)$, this choice does not seem controversial, as here D and P are very natural parameters, and the only ones in these simple cases. However, for the two scenarios with incoherent compasses and with the presence of obstacles, there are several other possible parameters, and their choice may

raise a doubt. As mentioned in the introduction, in these two scenarios, distinct labels of agents are necessary to break symmetry, since rendezvous is impossible for anonymous agents. Hence any rendezvous algorithm has to use labels l and L as inputs, and thus the choice of these labels as parameters seems natural. By contrast, the choice of parameter x may seem more controversial. Why do we want to express the cost of a rendezvous algorithm in terms of the largest perimeter of an obstacle? Are there other natural choices of parameter sets? What are their implications?

Let us start by pondering the second question. It is not hard to give examples of other natural choices of parameters for the two scenarios with incoherent compasses and with the presence of obstacles. For example, in the hardest scenario (without a map), we could drop parameter x and try to express the cost of the same Algorithm RVO only in terms of D, P, l, and L. Since $x \leq P$, we would get $O(P|L|)$ instead of $O(P + x|L|)$. Incidentally, as in our lower bound example of terrains we have $x = \Theta(P)$, this new complexity $O(P|L|)$ is optimal for the same reason as the former one.

Another possibility would be adding, instead of dropping a parameter. We could, for example, add the parameter P_e which is the length of the external perimeter of the terrain, i.e., the perimeter of polygon $\mathcal{P}_0$. Then it becomes natural to modify Algorithm RVO as follows. The first two phases are the same. In the third phase, the agent goes around obstacle $\mathcal{O}$ and compares its perimeter to P_e. If the perimeter of $\mathcal{O}$ is smaller (or equal), then the algorithm proceeds as before, and if it is larger, then the agent goes back to the boundary of $\mathcal{P}_0$ and executes Phase 3 on this boundary instead of the boundary of $\mathcal{O}$. The new algorithm has complexity $O(P + \min(x, P_e)|L|)$. Its complexity is again optimal because in our lower bound example we can choose the parameter $y = \min(x, P_e)$ and enlarge the largest of the two boundaries by lengthy but thin zigzags. Thus we can preserve the lower bound $\Omega(P + \min(x, P_e)|L|)$, even when x and P_e differ significantly.

The reason why we chose parameters D, P, l, L, and x instead of just D, P, l and L, is that complexity $O(P + x|L|)$ shows a certain continuity of the complexity of Algorithm RVO with respect to the sizes of obstacles: when the largest obstacle decreases, this complexity approaches $O(P)$ and it becomes $O(P)$ if there are no obstacles. In this case our algorithm coincides with Algorithm RV. This is not the case with complexity $O(P|L|)$. On the other hand, this choice coincides with $O(P + \min(x, P_e)|L|)$ in many important cases, for example for convex obstacles (as then we have $x < P_e$).

It is then natural to ask what happens if we add parameter x in the scenario with incoherent compasses and with the presence of obstacles but with the map. Obviously we could still use Algorithm RVO and get complexity $O(P + x|L|)$. However, our lower bound argument in this scenario gives in fact only $\Omega(D + \min(x, D)|l|)$. In our example we had $D = \Theta(x)$ but we only get $\Omega(D + x|l|)$ even if D is much larger than x. On the other hand, if D is much smaller than x we can only get the lower bound $\Omega(D|l|)$ because it matches the complexity of $RVMO$ in this case. Hence it is natural to ask if there exists a rendezvous algorithm with cost $O(D + \min(x, D)|l|)$ for arbitrary terrains in this scenario. We leave this as an open question.

References

[1] R. Aleliunas, R.M. Karp, R.J. Lipton, L. Lovász, and C. Rackoff, Random walks, universal traversal sequences, and the complexity of maze problems, Proc. Annual Symposium on Foundations of Computer Science FOCS'1979, 218-223.

[2] S. Alpern, The rendezvous search problem, SIAM J. on Control and Optimization 33 (1995), 673-683.

[3] S. Alpern, Rendezvous search on labelled networks, Naval Reaserch Logistics 49 (2002), 256-274.

[4] S. Alpern and S. Gal, The theory of search games and rendezvous. Int. Series in Operations research and Management Science, number 55, Kluwer Academic Publishers, 2002. Kluwer Academic Publisher, 2002.

[5] J. Alpern, V. Baston, and S. Essegaier, Rendezvous search on a graph, Journal of Applied Probability 36 (1999), 223-231.

[6] E. Anderson and R. Weber, The rendezvous problem on discrete locations, Journal of Applied Probability 28 (1990), 839-851.

[7] E. Anderson and S. Fekete, Asymmetric rendezvous on the plane, Proc. 14th Annual ACM Symp. on Computational Geometry, 1998.

[8] E. Anderson and S. Fekete, Two-dimensional rendezvous search, Operations Research 49 (2001), 107-118.

[9] E. M. Arkin, J. S. B. Mitchell and C. D. Piatko, Bicriteria shortest path problems in the plane, In Proc. 3rd Canad. Conf. Comput. Geom, 153-156, 1991.

[10] V. Baston and S. Gal, Rendezvous on the line when the players' initial distance is given by an unknown probability distribution, SIAM J. on Control and Optimization 36 (1998), 1880-1889.

[11] V. Baston and S. Gal, Rendezvous search when marks are left at the starting points, Naval Res. Log. 48 (2001), 722-731.

[12] H. Blum, A transformation for extracting new descriptors of shape. In W. Whaten-Dunn, Proc. Symp. Models for Perception of Speech and Visual Form, 362-380. MIT Press, 1967.

[13] M. Cieliebak, P. Flocchini, G. Prencipe, N. Santoro, Solving the Robots Gathering Problem, Proc. 30th International Colloquium on Automata, Languages and Programming (ICALP 2003), 1181-1196.

[14] G. De Marco, L. Gargano, E. Kranakis, D. Krizanc, A. Pelc, U. Vaccaro, Asynchronous deterministic rendezvous in graphs, Theoretical Computer Science 355 (2006), 315-326.

[15] D. Coppersmith, P. Doyle, P. Raghavan, and M. Snir, Random walks on weighted graphs, and applications to on-line algorithms, Proc. 22nd Annual ACM Symposium on Theory of Computing (STOC'1990), 369-378.

[16] F. Chin, J. Snoeyink, and C. A. Wang, Finding the Medial Axis of a Simple Polygon in Linear Time, Discrete Comput. Geom., 382–391, 1995.

[17] A. Dessmark, P. Fraigniaud, D. Kowalski, and A. Pelc, Deterministic rendezvous in graphs, Algorithmica 46 (2006), 69-96.

[18] M. Dyer, A. Frieze, and R. Kannan, A random polynomial time algorithm for estimating volumes of convex bodies, Proc. 21st Annual ACM Symposium on Theory of Computing (STOC'1989), 375-381.

[19] P. Flocchini, G. Prencipe, N. Santoro, P. Widmayer, Gathering of asynchronous oblivious robots with limited visibility, Proc. 18th Annual Symposium on Theoretical Aspects of Computer Science STACS'2001, LNCS 2010, 247-258.

[20] S. Gal, Rendezvous search on the line, Operations Research 47 (1999), 974-976.

[21] J. Hershberger and S. Suri, An Optimal Algorithm for Euclidean Shortest Paths in the Plane, SIAM J. Comput. 28, 2215-2256, 1997.

[22] A. Israeli and M. Jalfon, Token management schemes and random walks yield self stabilizing mutual exclusion, Proc. PODC'1990, 119-131.

[23] R. Klasing, A. Kosowski, A. Navarra, Taking advantage of symmetries: gathering of asynchronous oblivious robots on a ring. Proc. 12th International Conference on Principles of Distributed Systems, (OPODIS 2008), 446-462.

[24] R. Klasing, E. Markou, A. Pelc, Gathering asynchronous oblivious mobile robots in a ring, Theoretical Computer Science 390 (2008), 27-39.

[25] D. Kowalski, A. Malinowski, How to meet in anonymous network. Theoretical Computer Science 399 (2008), 141-156.

[26] E. Kranakis, D. Krizanc, N. Santoro and C. Sawchuk, Mobile agent rendezvous in a ring, Proc. 23rd International Conference on Distributed Computing Systems (ICDCS'2003), 592-599.

[27] W. Lim and S. Alpern, Minimax rendezvous on the line, SIAM J. on Control and Optimization 34 (1996), 1650-1665.

[28] G. Prencipe, Impossibility of gathering by a set of autonomous mobile robots, Theoretical Computer Science 384 (2007), 222-231.

[29] F. P. Preperata, The medial axis of a simple polygon, Mathematical Foundations of Computer Science (LNCS), 53:443-450,1977

[30] T. Schelling, The strategy of conflict, Oxford University Press, Oxford, 1960.

[31] G. Stachowiak, Asynchronous Deterministic Rendezvous on the Line, SOFSEM 2009: 497-508.

[32] L. Thomas, Finding your kids when they are lost, Journal on Operational Res. Soc. 43 (1992), 637-639.

[33] X. Yu and M. Yung, Agent rendezvous: a dynamic symmetry-breaking problem, Proc. International Colloquium on Automata, Languages, and Programming (ICALP'1996), LNCS 1099, 610-621.

[34] A. Ta-Shma, U. Zwick, Deterministic rendezvous, treasure hunts and strongly universal exploration sequences., Proc. 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2007), 599-608.